

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant : Nigel Green

Confirmation No. 5376

Appl. No. : 10/720,712

Filed : November 24, 2003

For : DEFERRED AND OFF-LOADED
RENDERING OF SELECTED
PORTIONS OF WEB PAGES TO
INCORPORATE LATE-ARRIVING
SERVICE DATA

Examiner : Lin Liu

APPEAL BRIEF

United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

This Appeal Brief is being filed in response to the final Office Action dated October 10, 2009 (the "Final Office Action").

I. REAL PARTY IN INTEREST

The real party in interest in the present application is Amazon Technologies, Inc., the assignee of the application.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are currently pending.

III. STATUS OF CLAIMS

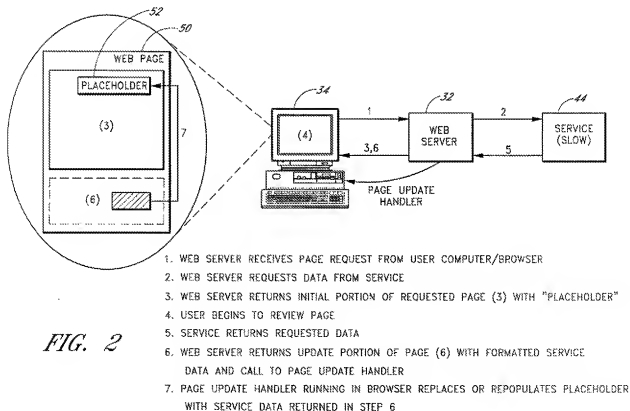
Claims 1-43 and 49-68, which are included in the appendix, are currently pending in the application and are the subject of this appeal. All of these claims stand rejected. Claims 44-48 are canceled.

IV. STATUS OF AMENDMENTS

No amendments were filed in response to the Final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application includes four independent claims – claim 1, claim 18, claim 27 and claim 52. Each independent claim is summarized below, with citations (in bold, bracketed text) to corresponding portions of the specification and drawings as required by 37 C.F.R. § 41.37(c)(1)(v). These citations illustrate specific examples and embodiments of the recited claim language, and do not limit the claims. Except where indicated otherwise, the cited reference numbers refer to events and other items shown in Figure 2, which is reproduced below.



Claim 1

Claim 1 is directed to a method of dynamically generating and serving web pages [see, e.g., page 2, ¶ 0005]. The method comprises at least the following:

- receiving a page request [event 1] at a server [32], the page request generated by a web browser [38 in Fig. 1] running on a user computer [34] and corresponding to a web page [50] that is generated dynamically [see, e.g., page 2, lines 6-10 of ¶ 0005; and page 6, first 2 lines of ¶ 0023];
- in response to the page request, sending a service request [event 2] from the server [32] to a service [44] to request service data to incorporate into the web page [see, e.g., page 6, lines 2-4 of ¶ 0023; Fig. 5, block 64; and page 14, last 3 lines of ¶ 0054];
- before the service returns the service data, transmitting a first portion [(3)] of the web page [event 3] from the server [32] to the user computer [34] for display by the web browser, said first portion [(3)] including viewable content that is viewable on the user computer [34] while the service request is pending [event 4], and including a placeholder [52] for the requested service data [see, e.g., page 2, last 8 lines of ¶ 0005; page 6, ¶ 0025; page 7, ¶ 0027 and first 3 lines of ¶ 0028; pages 9-10, ¶ 0037; and Fig. 5, block 68];
- after the service returns the service data [event 5] and before the web page has been fully loaded, transmitting from the server [32] to the web browser a second portion [(6)] of the web page [event 6], the second portion including the service data [see, e.g., page 2, first 3 lines of ¶ 0006; page 7, ¶ 0028 and first 8 lines of ¶ 0029; page 9, ¶ 0034; and Fig. 5, block 72]; and
- transmitting to the user computer [34] a page update handler [see arrow labeled “page update handler” in Fig. 2] which, when executed by the web browser [event 7], incorporates the service data included within the second portion [(6)] of the web page [50] into the first portion [(3)] of the web page in a viewable form [see, e.g., page 2, first 6 lines of ¶ 0005; page 2, first 3 lines of ¶ 0006; page 6, ¶¶ 0019-21; last 2 lines of ¶ 0029 on pages 7 and 8; and page 8, ¶ 0030].

Claim 18

Claim 18 is directed to a method of responding to a request from a web browser for a web page. The method comprises at least the following:

- sending a service request [event 2] to a service [44] to request service data to be displayed within a portion [(3)] of the web page [see, e.g., page 6, lines 2-4 of ¶ 0023; Fig. 5, block 64; and page 14, last 3 lines of ¶ 0054];
- if the service [44] returns the requested service data within a selected time interval, populating said portion [(3)] of the web page [50] with the service data prior to transmitting the web page to the web browser [see, e.g., page 2, first 5 lines of ¶ 0007; page 10, first 11 lines of ¶ 0040; Fig. 5, blocks 74-78; and page 15, first 5 lines of ¶ 0057] ; and
- if the service [44] does not return the requested service data within the selected time interval: (a) transmitting at least said portion [(3)] of the web page [50] to the web browser without the service data while the service request is pending [event 3], (b) in response to receiving the requested service data from the service [44], transmitting the service data to the web browser [events 5 and 6], and (c) invoking a page update handler [event 6] which, when executed by the web browser [event 7], populates said portion [(3)] of the web page [50] with the service data transmitted in (b) [see, e.g., page 2, ¶¶ 0005-0007; page 6, ¶¶ 0025-0030; Fig. 5, blocks 78 and 68-72; and page 15, last 4 lines of ¶ 0057];
- wherein the method is performed by a server system that comprises one or more physical servers [see, e.g. page 2, ¶ 0005; and pages 3 and 4, ¶ 0015]

Claim 27

Claim 27 is directed to a method of generating a web page [50] in response to a request from a web browser, the method comprising:

- (a) sending a service request [event 2] to a service [44] to request service data to be displayed in the web page [see, e.g., page 6, lines 2-4 of ¶ 0023; Fig. 5, block 64; and page 14, last 3 lines of ¶ 0054];

(b) transmitting to the web browser at least a first portion [(3)] of the web page [event 3], said first portion [(3)] including content that is viewable within the web browser [event 4] while the service request is pending [see, e.g., page 2, last 8 lines of ¶ 0005; page 6, ¶ 0025; page 7, ¶ 0027 and first 3 lines of ¶ 0028; pages 9-10, ¶ 0037; and Fig. 5, block 68];

(c) after the service [44] responds to the service request by returning the service data [event 5], sending the service data [event 6] to the web browser [see, e.g., page 2, first 3 lines of ¶ 0006; and page 7, ¶ 0028]; and

(d) instructing the web browser to execute a page update handler that, when executed [event 7], incorporates a viewable representation of the service data, as transmitted in (c), into the first portion [(3)] of the web page [see, e.g., page 2, first 6 lines of ¶ 0005; page 6, ¶ 0019; and page 8, ¶ 0030];

wherein the method, including (a), (b), (c) and (d), is performed by a web server system that comprises one or more physical servers [see, e.g. page 2, ¶ 0005; and pages 3 and 4, ¶ 0015].

Claim 52

Claim 52 is directed to a system for responding to web page requests. The system comprises at least the following:

- a web server system that comprises one or more physical servers [see, e.g. page 2, ¶ 0005; and pages 3 and 4, ¶ 0015], said web server system responsive to page requests from browsers [38 in Fig. 1] running on user computing devices [34] by generating and serving web pages [50] that include data retrieved from one or more services [44], said web server system operative to respond to a request from a browser for a web page according to a process that comprises:
 - sending a service request [event 2] to a service [44] to request service data to be displayed in the web page [see, e.g., page 6, lines 2-4 of ¶ 0023; Fig. 5, block 64; and page 14, last 3 lines of ¶ 0054];
 - transmitting to the browser at least a first portion [(3)] of the web page [event 3], said first portion including content that is viewable with the browser while the

service request is pending [*see, e.g.*, page 2, last 8 lines of ¶ 0005; page 6, ¶ 0025; page 7, ¶ 0027 and first 3 lines of ¶ 0028; pages 9-10, ¶ 0037; and Fig. 5, block 68];

- after the service responds to the service request by returning the service data [event 5], sending the service data [event 6] to the browser [*see, e.g.*, page 2, first 3 lines of ¶ 0006; and page 7, ¶ 0028]; and
- causing the browser to execute a page update handler [events 6 and 7] that, when executed, causes a viewable representation of the service data to be incorporated into the first portion [(3)] of the web page [*see, e.g.*, page 2, first 6 lines of ¶ 0005; page 6, ¶ 0019; and page 8, ¶ 0030].

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following issues are presented on appeal:

1. Whether claims 1-6, 8-14, 16-37, 39-43, 49-60 and 62-68 are properly rejected under 35 U.S.C. § 102(e) over Mateos (U.S. Pub. 2003/0050995).
2. Whether claim 7 is properly rejected under 35 U.S.C. § 103(a) over Mateos in view of Starkey (U.S. Pub. 2002/0059327).
3. Whether claims 15, 38 and 61 are properly rejected under 35 U.S.C. § 103(a) over Mateos in view of Samar (U.S. Pat. 6,563,514).

VII. ARGUMENT

For the reasons explained below, the rejections under §§ 102 and 103 are improper. By declining to separately argue certain dependent claims, Appellant does not imply that the features recited in these claims are taught or suggested by the cited art. In addition, no admission is made that any of the references constitutes “prior art.”

1. Claims 1-6, 8-14, 16-37, 39-43, 49-60 and 62-68 are not anticipated by Mateos.

Claims 1-6, 8-14, 16-37, 39-43, 49-60 and 62-68 stand rejected as anticipated by Mateos. All of the pending independent claims are included in this group.

“A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union*

Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). “Because the hallmark of anticipation is prior invention, the prior art reference—in order to anticipate under 35 U.S.C. § 102—must not only disclose all elements of the claim within the four corners of the document, but must also disclose those elements ‘arranged as in the claim.’” Net Moneyin v. Verisign, Inc. No. 07-1565 at 15 (Fed. Cir. Oct. 20, 2008), citing Connell v. Sears, Roebuck & Co., 722 F.2d 1542, 1548 (Fed. Cir. 1983).

As will be explained below, the rejection is improper because Mateos does not expressly or inherently disclose all of the elements of any claim recited in the present application. Each independent claim and several dependent claims are discussed below.

Mateos

Mateos discloses a process in which a server incorporates “dynamic information” retrieved from a database into a requested web page together with executable instructions for formatting and displaying this data. Once the page is received by the web browser/user computer, the executable instructions cause this dynamic information to be properly displayed on the page. The executable instructions form part of a “view template” that is sent to the client computer with the dynamic information. *See, e.g.*, abstract and paragraphs 0073 and 0076 of Mateos.

Mateos never discloses or suggests that the requested web page, or any portion of it, is or can be sent to the client computer before all of the dynamic information is retrieved from the database. To the contrary, the transmission of the page in Mateos does not begin until after the last database record has been retrieved. *See, e.g.*, Fig. 4, blocks 445-455 and paragraphs 0070-0073 of Mateos. Thus, in contrast to the process disclosed in the present application, the user cannot begin to review any portion of the requested page until after the server has retrieved all of the dynamic information from the database. For example, if the database in Mateos takes ten seconds to respond to a query, the user will experience a delay of at least ten seconds before the requested page begins to appear.

Independent Claim 1

Claim 1 reads as follows, which emphasis (*italics*) added for purposes of discussion:

A method of dynamically generating and serving web pages, the method comprising:

receiving a page request at a server, the page request generated by a web browser running on a user computer and corresponding to a web page that is generated dynamically;

in response to the page request, sending a service request from the server to a service to request service data to incorporate into the web page;

before the service returns the service data, transmitting a first portion of the web page from the server to the user computer for display by the web browser, said first portion including viewable content that is viewable on the user computer while the service request is pending, and including a placeholder for the requested service data;

after the service returns the service data and before the web page has been fully loaded, transmitting from the server to the web browser a second portion of the web page, the second portion including the service data; and

transmitting to the user computer a page update handler which, when executed by the web browser, incorporates the service data included within the second portion of the web page into the first portion of the web page in a viewable form.

The rejection of claim 1 is improper at least because Mateos does not explicitly or inherently disclose the italicized portion of the claim in the context of the claim's other recitations. In connection with the italicized portion, the Final Office Action points to paragraphs 15, 58, 76 and 80 of Mateos. Final Office Action at page 3, first full paragraph. Neither these nor any other portion of Mateos, however, discloses the feature at issue. In this regard, contrary to the assertions in the Final Office Action, nowhere does Mateos ever suggest that the disclosed "view template" can be sent to and displayed on the client computer before the server has finished retrieving the corresponding dynamic information. Each referenced paragraph is discussed below.

Paragraph 15 of Mateos explains that the server retrieves dynamic information and a corresponding view template, and sends both to the client computer. The client computer then combines the dynamic information with the view template to display the dynamic information

according to the view template. Nothing in this paragraph suggests that the view template is sent to the client computer before the server has retrieved the dynamic information. In addition, nothing in this paragraph suggests that the view template is capable of being displayed on the client computer in the absence of the dynamic information.

Paragraph 0058 of Mateos similarly fails to suggest that the view template (or any other portion of the page) is sent to the client computer prior to the server's retrieval of the dynamic information. To the contrary, if read literally, this paragraph appears to state that the dynamic information is sent to the client computer *before* any other portion of the page.

Paragraph 0076 states that "the [database] records are sequentially fetched and a corresponding statement assigning the current record to a corresponding variable is inserted into a first section of the view template; [and that] a second section of the view template includes instructions for displaying the variables, which are executed on the client computer." This statement appears to indicate that the server populates the view template with variables as corresponding database records (i.e., the dynamic information) are retrieved, and that the view template therefore is not ready to be sent to the client computer until after the server has finished retrieving the dynamic information. (*See also* paragraphs 0070 and 0075, which explain how the view template is constructed during the retrieval of database records.) Thus, this paragraph similarly fails to suggest that the view template is sent to the client computer before the server has finished retrieving the dynamic information.

Paragraph 80 of Mateos reads as follows, with emphasis added for purposes of discussion:

[0080] Alternatively, the view template and the resulting web page have a different structure (for example the dynamic information is stored in a *separate file*), the method is implemented in a network that employs different structures to be displayed on the client computers, *the dynamic information and the corresponding instructions are transmitted to the client computer independently*, the dynamic information and the view template are combined in a different manner on the client computer, and so on.

Nothing in this paragraph suggests that the view template, or any other portion of the web page, is or can be sent to and displayed on the client computer before the server retrieves the page's dynamic information. In this regard, as suggested by the reference to "a separate file," the statement "the dynamic information and the corresponding instructions are transmitted to the

client computer *independently*” apparently means only that two can be sent *as separate files* once the dynamic information has been retrieved. Indeed, paragraph 0080 says nothing about the timing with which the view template is transmitted to the client computer.

Thus, the referenced portions of Mateos do not suggest, much less disclose, sending the view template or any other portion of the page to the client computer before the server has retrieved the page’s dynamic information.

In connection with this issue, the Final Office Action disregards Fig. 4, blocks 445-455 and paragraphs 0070-0073 of Mateos. These portions of Mateos make clear that the transmission of the page in Mateos does not begin until *after* the last database record has been retrieved. Nothing in Mateos suggests any other mode of operation.

In response to Appellant’s arguments, the Final Office Action includes the following rebuttal:

In response to applicant's argument **a**, the examiner disagrees. Mateo specifically discloses that the dynamic information and the view template are transmitted from the server computer, and the server computer does not need to have the entire dynamic information available on the server computer (Mateo: page 5, paragraphs 73-74) for viewing the dynamic information on client's computer. Additionally, Mateo discloses that the dynamic information and the corresponding instructions are transmitted to the client computer independently; the corresponding instruction is to be executed and populated the dynamic information with the view template to be update the information for the client (Mateo: page 2, paragraph 28 and page 5, paragraph 80). Therefore, it'd have been reasonable for a person of ordinary skill in the art at the time of invention to realize that the view template would have to be display to the client first before the dynamic information can be populated with the corresponding instructions. (Final Office Action at pages 1 and 11.)

This rebuttal disregards Fig. 4, blocks 445-455 and paragraphs 0070 and 0076 of Mateos. As explained above, these portions of Mateos make clear that (1) the transmission of the web page does not begin until the server has finished retrieving the dynamic information, and (2) the page’s view template is constructed as the dynamic information is retrieved, and thus is not yet ready to be transmitted. This rebuttal also erroneously assumes that paragraph 0080, through its use of the term “independently,” discloses or suggests the transmission and display of the view template before the server has retrieved the associated dynamic information. As explained above, it does not.

The other portions of Mateos cited in the above rebuttal (namely paragraphs 28, 73 and 74) similarly fail to support the rejection. Paragraph 28 explains the structure of the web server and the view templates, and says nothing about the timing with which the view templates are transmitted to the user/client computer. The reference to “updating” in the last sentence of this paragraph refers to the updating *of the web server* with new or revised pages and templates (see Fig. 2), and not to the updating of a web page as displayed on a client/user computer. Paragraph 0073 describes the overall process used by Mateos; the third and fourth sentences of this paragraph make clear that the view template is sent to the client computer *after* the dynamic information has been retrieved. Finally, paragraph 0074 merely explains how the disclosed process allegedly improves the server’s performance by enabling the retrieved database records to be processed on the client computers, rather than the server, to generate the viewable dynamic information. (See paragraphs 0004-0006 and 0075 of Mateos for additional information regarding the performance issues being addressed.) As with the other referenced portions of Mateos, nothing in paragraph 0074 suggests that the view template, or any other portion of the page, is sent to or displayed on the client computer before the corresponding database records are retrieved.

For at least these reasons, Mateos does not disclose the italicized portion of claim 1, and the anticipation rejection is improper as applied to claim 1 and the corresponding dependent claims.

Dependent Claim 8

The rejection of claim 8 is improper in view of the claim’s dependency from claim 1. In addition, the rejection of claim 8 is improper because Mateos does not disclose the following feature recited in claim 8: “the placeholder for the requested service data is included within the first portion of the web page in response to a failure of the service to return the service data within a selected time interval.” The Final Office Action points to paragraph 0028 of Mateos in support of rejecting claim 8. Final Office Action at page 4. The referenced paragraph, however, merely describes the web server and view templates used, and contains no disclosure whatsoever of the claimed feature.

Dependent Claim 9

The rejection of claim 9 is improper in view of the claim's dependency from claim 1. In addition, the rejection of claim 9 is improper because Mateos does not disclose the following feature recited in claim 9: "the placeholder for the requested service data is included within the first portion of the web page in response to a server decision to defer rendering of a portion of the web page, said server decision being based at least in part on response time data collected for the service." The Final Office Action points to paragraph 0075 of Mateos in support of the rejection of claim 9. Final Office Action at pages 4 and 5. The referenced paragraph, however, contains no disclosure whatsoever of a "server decision to defer rendering of a portion of the web page," much less such a decision that is "based at least in part on response time data collected for the service."

Dependent Claim 10

The rejection of claim 10 is improper in view of the claim's dependency from claim 1. In addition, the rejection of claim 10 is improper because Mateos does not disclose the following feature recited in claim 10: "the placeholder for the requested service data is included within the first portion of the web page in response to a server decision to defer rendering of a portion of the web page, said server decision taking into consideration at least one of the following: (a) a load level of the service, (b) a load level of a web server system that responds to the page request." The Final Office Action points to paragraph 0075 of Mateos in support of the rejection of claim 10. Final Office Action at page 5. The referenced paragraph, however, contains no disclosure whatsoever of the feature at issue.

Dependent Claim 16

The rejection of claim 16 is improper in view of the claim's dependency from claim 1. In addition, the rejection of claim 16 is improper because Mateos does not disclose the following feature recited in claim 16: "the page update handler selects a display format to use to display the service data in the web page based at least in part on a dimension of a window of the web browser running on the user computer." The Final Office Action points to paragraphs 0076-0079 of Mateos in support of the rejection of claim 16. Final Office Action at page 6. The referenced paragraphs, however, simply do not disclose or suggest the recited feature.

Dependent Claim 68

The rejection of claim 68 is improper in view of the claim's dependency from claim 1 (via intervening claim 67). In addition, the rejection of claim 68 is improper because Mateos does not disclose the following feature recited in claim 68: "wherein dynamically generating the first portion of the web page comprises incorporating into the web page a viewable status message reflecting that additional data is being retrieved." The Final Office Action points to paragraphs 0028, 0073 and 0074 of Mateos in support of the rejection of claim 68. Final Office Action at page 7. The referenced paragraphs, however, do not disclose the incorporation into a web page of a viewable status message as claimed.

Independent Claim 18

Claim 18 reads as follows:

A method of responding to a request from a web browser for a web page, the method comprising:

 sending a service request to a service to request service data to be displayed within a portion of the web page;

 if the service returns the requested service data within a selected time interval, populating said portion of the web page with the service data prior to transmitting the web page to the web browser; and

if the service does not return the requested service data within the selected time interval: (a) transmitting at least said portion of the web page to the web browser without the service data while the service request is pending, (b) in response to receiving the requested service data from the service, transmitting the service data to the web browser, and (c) invoking a page update handler which, when executed by the web browser, populates said portion of the web page with the service data transmitted in (b);

 wherein the method is performed by a server system that comprises one or more physical servers.

The rejection of claim 18 is improper at least because Mateos does not disclose the italicized portion of the claim, particularly when the italicized subparagraph is read in conjunction with the immediately preceding subparagraph.

The Final Office Action merely refers to the analysis of claims 1-6, 8-14, 16, 17 and 49 in support of rejecting claim 18. Final Office Action at page 6. As explained above, however, the rejection of claim 1 (and thus its dependent claims) is based on a misreading of Mateos. For

example, as explained above in connection with claim 1, Mateos never suggests sending the view template (or any other portion of the requested page) to the user computer/browser before the dynamic information has been fully retrieved. Thus, Mateos does not disclose element “a” in the italicized portion of claim 18.

Further, Mateos does not disclose a process in which the type or content of the page returned depends on whether the service returns “the requested service data within [a] selected time interval” as recited in claim 18. The portion of Mateos applied to dependent claim 8 (discussed above) simply does not disclose this feature in claim 18.

For at least these reasons, the anticipation rejection is improper as applied to claim 18 and the corresponding dependent claims.

Dependent Claim 21

The rejection of claim 21 is improper in view of the claim’s dependency from claim 18. In addition, the rejection of claim 21 is improper because Mateos does not disclose the following feature recited in claim 21: “the service data is transmitted to the web browser within a separate web page within a hidden window.” The Final Office Action effectively disregards claim 21 by asserting that its limitations are substantially the same as those of other claims addressed elsewhere. *See* Final Office Action at page 6.

Dependent Claim 24

The rejection of claim 24 is improper in view of the claim’s dependency from claim 18. In addition, the rejection of claim 24 is improper because Mateos does not disclose the following feature recited in claim 24: “the page update handler selects a display format to use to display the service data in the web page based at least in part on a size of a window of the web browser.” Neither paragraphs 0076-0079 nor any other portions of Mateos disclose this feature.

Independent claim 27

Claim 27 reads as follows:

A method of generating a web page in response to a request from a web browser, the method comprising:

(a) sending a service request to a service to request service data to be displayed in the web page;

(b) *transmitting to the web browser at least a first portion of the web page, said first portion including content that is viewable within the web browser while the service request is pending;*

(c) after the service responds to the service request by returning the service data, sending the service data to the web browser; and

(d) instructing the web browser to execute a page update handler that, when executed, incorporates a viewable representation of the service data, as transmitted in (c), into the first portion of the web page;

wherein the method, including (a), (b), (c) and (d), is performed by a web server system that comprises one or more physical servers.

The rejection of claim 27 is improper at least because Mateos does not disclose the italicized portion of the claim in the context of the claim's other recitations.

In connection with this claim, the Final Office Action merely refers to the analysis of claims 1-6, 8-14, 16, 17 and 49. Final Office Action at page 6, last paragraph. As explained above, however, the rejection of claim 1 (and thus its dependent claims) is based on a misreading of Mateos. For example, as explained above, Mateos never suggests sending the view template (or any other portion of the requested page) to the user computer/browser before the dynamic information has been fully retrieved. Thus, Mateos does not disclose "transmitting to the web browser at least a first portion of the web page, said first portion including content that is viewable within the web browser while the service request is pending" as recited in claim 27.

For at least these reasons, the anticipation rejection is improper as applied to claim 27 and the corresponding dependent claims.

Dependent Claim 28

The rejection of claim 28 is improper in view of the claim's dependency from claim 27. In addition, the rejection of claim 28 is improper because Mateos does not disclose the following feature recited in claim 28: "the first portion of the web page includes a display object that is

initially displayed without the service data, and which is subsequently populated with the service data by the page update handler.” The Final Office Action effectively disregards portions of claim 28 (and particularly “initially displayed without the service data”) by asserting that the claim’s limitations are substantially the same as those of claims 1-6, 8-14, 16, 17 and 49. *See* Final Office Action at page 6, last paragraph.

Dependent Claim 31

The rejection of claim 31 is improper in view of the claim’s dependency from claim 27. In addition, the rejection of claim 31 is improper because Mateos does not disclose the following feature recited in claim 31: “step (c) comprises transmitting the service data to the web browser as part of a secondary web page that is loaded by the web browser within a hidden window and is accessed by the page update handler.” The Final Office Action effectively disregards claim 31 by asserting that the claim’s limitations are substantially the same as those of claims 1-6, 8-14, 16, 17 and 49. *See* Final Office Action at page 6, last paragraph.

Dependent Claim 35

The rejection of claim 35 is improper in view of the claim’s dependency from claim 27. In addition, the rejection of claim 35 is improper because Mateos does not disclose the following feature recited in claim 35: “at least step (d) is performed in response to a failure of the service to return the requested service data within a selected time interval.” In connection with this claim, the Final Office Action appears to rely on paragraph 0028 of Mateos, which was also applied to claim 8. Final Office Action at page 6, last paragraph. Paragraph 0028, however, merely describes the web server and view templates used, and contains no disclosure whatsoever of the claimed feature.

Dependent Claim 36

The rejection of claim 36 is improper in view of the claim’s dependency from claim 27. In addition, the rejection of claim 36 is improper because Mateos does not disclose the following feature recited in claim 36: “at least step (d) is performed in response to a server decision to defer rendering of a portion of the web page, said server decision being based at least in part on response time data collected for the service.” In connection with this claim, the Final Office Action appears to rely on paragraph 0075 of Mateos, which was also applied to claim 9. Final Office Action at page 6, last paragraph. Paragraph 0075, however, contains no disclosure

whatsoever of “a server decision to defer rendering of a portion of the web page,” much less such a server decision that is “based at least in part on response time data collected for the service.”

Dependent Claim 42

The rejection of claim 42 is improper in view of the claim’s dependency from claim 27. In addition, the rejection of claim 42 is improper because Mateos does not disclose the following feature recited in claim 42: “the page update handler selects a display format to use to display the service data in the web page based at least in part on a dimension of a window of the web browser.” Neither paragraphs 0076-0070 nor any other portions of Mateos disclose this feature.

Independent claim 52

Claim 52 reads as follows:

A system for responding to web page requests, the system comprising:

a web server system that comprises one or more physical servers, said web server system responsive to page requests from browsers running on user computing devices by generating and serving web pages that include data retrieved from one or more services, said web server system operative to respond to a request from a browser for a web page according to a process that comprises:

sending a service request to a service to request service data to be displayed in the web page;

transmitting to the browser at least a first portion of the web page, said first portion including content that is viewable with the browser while the service request is pending;

after the service responds to the service request by returning the service data, sending the service data to the browser; and

causing the browser to execute a page update handler that, when executed, causes a viewable representation of the service data to be incorporated into the first portion of the web page.

The rejection of claim 52 is improper at least because Mateos does not disclose the italicized portion of the claim in the context of the claim’s other recitations.

In connection with this claim, the Final Office Action merely refers to the analysis of claims 1-6, 8-14, 16, 17 and 49. Final Office Action at page 7, first paragraph. As explained above, however, the rejection of claim 1 (and its dependent claims) is based on a misreading of Mateos. For example, as explained above, Mateos never suggests sending the view template (or

any other portion of the requested page) to the user computer/browser before the dynamic information has been fully retrieved. Thus, Mateos does not disclose “transmitting to the browser at least a first portion of the web page, said first portion including content that is viewable with the browser while the service request is pending” as recited in claim 52.

For at least these reasons, the anticipation rejection is improper as applied to claim 52 and the corresponding dependent claims.

Dependent Claim 53

The rejection of claim 53 is improper in view of the claim’s dependency from claim 52. In addition, the rejection of claim 53 is improper because Mateos does not disclose the following feature recited in claim 53: “[the] web server system is operative to cause the browser (a) to initially display a display object on the web page without the service data, and (b) to subsequently populate the display object with the service data via execution of the page update handler.” In connection with this claim, the Final Office Action refers to the analysis of claims 1-6, 8-14, 16, 17 and 49. Final Office Action at page 7, first paragraph. None of the portions of Mateos applied to claims 1-6, 8-14, 16, 17 and 49, however, discloses this feature recited in claim 53.

Dependent Claim 55

The rejection of claim 55 is improper in view of the claim’s dependency from claim 52. In addition, the rejection of claim 55 is improper because Mateos does not disclose the following feature recited in claim 55: “the web server system is operative to send the service data as part of a secondary web page that is accessed by the page update handler.” The Final Office Action, in support of rejecting claim 55, refers to the analysis of claims 1-6, 8-14, 16, 17 and 49. Final Office Action at page 7, first paragraph. None of the portions of Mateos applied to claims 1-6, 8-14, 16, 17 and 49, however, discloses this feature recited in claim 55.

Dependent Claim 59

The rejection of claim 59 is improper in view of the claim’s dependency from claim 52. In addition, the rejection of claim 59 is improper because Mateos does not disclose the following feature recited in claim 59: “the web server system is operative to selectively invoke said process based on a programmatic determination that is dependent upon a response time of said service.” In connection with this claim, the Final Office Action refers to the analysis of claims 1-6, 8-14,

16, 17 and 49. Final Office Action at page 7, first paragraph. None of the portions of Mateos applied to claims 1-6, 8-14, 16, 17 and 49, however, discloses this feature recited in claim 59.

Dependent Claim 63

The rejection of claim 63 is improper in view of the claim's indirect dependency from claim 52. In addition, the rejection of claim 63 is improper because Mateos does not disclose the following feature recited in claim 63: "the page update handler comprises executable code for selecting, based at least in part on a dimension of a window of the browser, a display format to use to display the service data in the web page." In connection with this claim, the Final Office Action refers to the analysis of claims 1-6, 8-14, 16, 17 and 49. Final Office Action at page 7, first paragraph. None of the portions of Mateos applied to claims 1-6, 8-14, 16, 17 and 49, however, discloses this feature recited in claim 63.

2. Claim 7 is improperly rejected over Mateos in view of Starkey.

Dependent claim 7 stands rejected under 35 U.S.C. § 103(a) over Mateos in view of Starkey. The rejection is improper at least because Mateos and Starkey do not collectively disclose or suggest the method of claim 1, from which claim 7 depends. In this regard, Starkey (the secondary reference applied to claim 7) does not overcome the above-noted deficiencies in Mateos as applied to claim 1, and the Final Office Action does not contend otherwise.

3. Claims 15, 38 and 61 are improperly rejected over Mateos in view of Samar.

Dependent claims 15, 38 and 61 stand rejected under 35 U.S.C. § 103(a) over Mateos in view of Samar. The rejections of these claims are improper at least because Mateos and Samar do not collectively teach or suggest the subject matter of independent claims 1, 27 and 52, from which claims 15, 38 and 61 respectively depend. In this regard, Samar (the secondary reference applied to claims 15, 38 and 61) does not overcome the above-noted deficiencies in Mateos as applied to claims 1, 27 and 52, and the Final Office Action does not contend otherwise.

Appl. No. : 10/720,712
Filed : November 24, 2003

VIII. CONCLUSION

For the reasons explained above, Appellant respectfully submits that the rejections of Claims 1-43 and 49-68 are improper and requests that these rejections be reversed.

Please charge any additional fees that may be required now or in the future to Deposit Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: December 17, 2009

By: /Ronald J. Schoenbaum/
Ronald J. Schoenbaum
Reg. No. 38,297
Customer No. 20,995
949-721-2950

CLAIMS APPENDIX

1. A method of dynamically generating and serving web pages, the method comprising:

receiving a page request at a server, the page request generated by a web browser running on a user computer and corresponding to a web page that is generated dynamically;

in response to the page request, sending a service request from the server to a service to request service data to incorporate into the web page;

before the service returns the service data, transmitting a first portion of the web page from the server to the user computer for display by the web browser, said first portion including viewable content that is viewable on the user computer while the service request is pending, and including a placeholder for the requested service data;

after the service returns the service data and before the web page has been fully loaded, transmitting from the server to the web browser a second portion of the web page, the second portion including the service data; and

transmitting to the user computer a page update handler which, when executed by the web browser, incorporates the service data included within the second portion of the web page into the first portion of the web page in a viewable form.

2. The method of Claim 1, wherein the placeholder comprises a display object, and the page update handler populates the display object with at least some of the service data included within the second portion of the web page.

3. The method of Claim 2, wherein the display object is positioned above at least some of said viewable content within the first portion of the web page.

4. The method of Claim 1, wherein the service data is included in the second portion of the web page in a condensed form in which at least some format coding is omitted, and the page update handler adds format coding to the service data to format the service data for display, whereby a quantity of data transmitted to the web browser is reduced.

5. The method of Claim 1, wherein the service data is included in the second portion of the web page in a hidden format.

6. The method of Claim 1, wherein the page update handler is transmitted to the user computer as part of the first portion of the web page.

7. The method of Claim 1, wherein the page update handler is transmitted to the user computer as part of a library file, separately from the web page.

8. The method of Claim 1, wherein the placeholder for the requested service data is included within the first portion of the web page in response to a failure of the service to return the service data within a selected time interval.

9. The method of Claim 1, wherein the placeholder for the requested service data is included within the first portion of the web page in response to a server decision to defer rendering of a portion of the web page, said server decision being based at least in part on response time data collected for the service.

10. The method of Claim 1, wherein the placeholder for the requested service data is included within the first portion of the web page in response to a server decision to defer rendering of a portion of the web page, said server decision taking into consideration at least one of the following: (a) a load level of the service, (b) a load level of a web server system that responds to the page request.

11. The method of Claim 1, wherein the second portion of the web page includes a command that causes the web browser to execute the page update handler.

12. The method of Claim 1, wherein the first portion of the web page includes a command that causes the web browser to execute the page update handler upon completion of loading of the web page.

13. The method of Claim 1, wherein the page update handler comprises a JavaScript function.

14. The method of Claim 1, wherein the service request is one of a plurality of service requests generated in response to the page request.

15. The method of Claim 1, wherein the page update handler incorporates the service data into the first portion of the web page as mouse-over text that is displayed by the web browser when a mouse cursor is positioned over a corresponding display element.

16. The method of Claim 1, wherein the page update handler selects a display format to use to display the service data in the web page based at least in part on a dimension of a window of the web browser running on the user computer.

17. The method of Claim 1, wherein the page update handler selects a display format to use to display the service data in the web page based at least in part on a quantity of the service data.

18. A method of responding to a request from a web browser for a web page, the method comprising:

 sending a service request to a service to request service data to be displayed within a portion of the web page;

 if the service returns the requested service data within a selected time interval, populating said portion of the web page with the service data prior to transmitting the web page to the web browser; and

 if the service does not return the requested service data within the selected time interval: (a) transmitting at least said portion of the web page to the web browser without the service data while the service request is pending, (b) in response to receiving the requested service data from the service, transmitting the service data to the web browser, and (c) invoking a page update handler which, when executed by the web browser, populates said portion of the web page with the service data transmitted in (b);

 wherein the method is performed by a server system that comprises one or more physical servers.

19. The method of Claim 18, wherein the page update handler incorporates the service data into the portion of the web page above other viewable content included in the portion of the web page.

20. The method of Claim 18, wherein the service data is transmitted to the web browser as part of the web page, and is moved or copied to the portion of the web page by the page update handler.

21. The method of Claim 18, wherein the service data is transmitted to the web browser within a separate web page within a hidden window.

22. The method of Claim 18, wherein invoking the page update handler comprises including, within the web page portion transmitted in (a), a command that causes the web browser to execute the page update handler upon completion of loading the web page.

23. The method of Claim 18, wherein the page update handler populates the portion of the web page with the service data before the web page has finished loading.

24. (Original) The method of Claim 18, wherein the page update handler selects a display format to use to display the service data in the web page based at least in part on a size of a window of the web browser.

25. The method of Claim 18, wherein the page update handler selects a display format to use to display the service data in the web page based at least in part on a quantity of the service data.

26. A web server system configured to respond to page requests from web browsers according to the method of Claim 18.

27. A method of generating a web page in response to a request from a web browser, the method comprising:

- (a) sending a service request to a service to request service data to be displayed in the web page;

- (b) transmitting to the web browser at least a first portion of the web page, said first portion including content that is viewable within the web browser while the service request is pending;

- (c) after the service responds to the service request by returning the service data, sending the service data to the web browser; and

- (d) instructing the web browser to execute a page update handler that, when executed, incorporates a viewable representation of the service data, as transmitted in (c), into the first portion of the web page;

wherein the method, including (a), (b), (c) and (d), is performed by a web server system that comprises one or more physical servers.

28. The method of Claim 27, wherein the first portion of the web page includes a display object that is initially displayed without the service data, and which is subsequently populated with the service data by the page update handler.

29. The method of Claim 28, wherein the display object is positioned above at least some of said content within the first portion of the web page.

30. The method of Claim 27, wherein step (c) comprises transmitting the service data as part of the web page before the web page has finished loading.

31. The method of Claim 27, wherein step (c) comprises transmitting the service data to the web browser as part of a secondary web page that is loaded by the web browser within a hidden window and is accessed by the page update handler.

32. The method of Claim 27, wherein the first portion of the web page is transmitted to the web browser while the service request is pending.

33. The method of Claim 27, wherein step (d) comprises including within the first portion of the web page a command that causes the web browser to execute the page update handler upon completion of loading the web page.

34. The method of Claim 27, wherein step (d) comprises including with the service data transmitted in step (c) a command that causes the web browser to execute the page update handler.

35. The method of Claim 27, wherein at least step (d) is performed in response to a failure of the service to return the requested service data within a selected time interval.

36. The method of Claim 27, wherein at least step (d) is performed in response to a server decision to defer rendering of a portion of the web page, said server decision being based at least in part on response time data collected for the service.

37. The method of Claim 27, wherein step (d) in response to a server decision to defer rendering of a portion of the web page, said server decision taking into consideration at least one of the following: (1) a load level of the service, (2) a load level of a web server system that responds to the request from the web browser.

38. The method of Claim 27, wherein the page update handler incorporates the service data into the first portion of the web page as mouse-over text that is displayed by the web browser when a mouse cursor is positioned over a corresponding display element.

39. A web server system configured to perform the method of Claim 27.
40. A web page generated according to the method of Claim 27 represented within a computer memory.
41. The method of Claim 27, wherein step (c) comprises sending the service data to the web browser in a substantially unformatted form to reduce a quantity of data transmitted to the web browser, wherein the page update handler adds format coding to the service data to incorporate the viewable representation of the service data into the web page.
42. The method of Claim 27, wherein the page update handler selects a display format to use to display the service data in the web page based at least in part on a dimension of a window of the web browser.
43. The method of Claim 27, wherein the page update handler selects a display format to use to display the service data in the web page based at least in part on a quantity of the service data.
- 44-48: (Canceled)
49. The method of Claim 1, wherein the method is performed by a web server system that comprises one or more physical servers.
50. The method of Claim 18, wherein the method additionally comprises the server system responding to said request from the web browser by dynamically generating said portion of the web page using data retrieved from at least one additional service.
51. The method of Claim 27, wherein the method additionally comprises the web server system responding to said request from the web browser by dynamically generating said first portion of the web page using data retrieved from at least one additional service.

52. A system for responding to web page requests, the system comprising:
a web server system that comprises one or more physical servers, said web server system responsive to page requests from browsers running on user computing devices by generating and serving web pages that include data retrieved from one or more services, said web server system operative to respond to a request from a browser for a web page according to a process that comprises:

 sending a service request to a service to request service data to be displayed in the web page;

 transmitting to the browser at least a first portion of the web page, said first portion including content that is viewable with the browser while the service request is pending;

 after the service responds to the service request by returning the service data, sending the service data to the browser; and

 causing the browser to execute a page update handler that, when executed, causes a viewable representation of the service data to be incorporated into the first portion of the web page.

53. The system of Claim 52, wherein web server system is operative to cause the browser (a) to initially display a display object on the web page without the service data, and (b) to subsequently populate the display object with the service data via execution of the page update handler.

54. The system of Claim 52, wherein the web server system is operative to send the service data to the browser as part of the web page before loading of the web page by the browser is complete.

55. The system of Claim 52, wherein the web server system is operative to send the service data as part of a secondary web page that is accessed by the page update handler.

56. The system of Claim 52, wherein the web server system is operative to send the first portion of the web page to the browser while the service request is pending.

57. The system of Claim 52, wherein the web server system is operative to cause the browser to execute the page update handler by including, within the first portion of the web page,

a command that instructs the browser to execute the page update handler upon completion of loading of the web page.

58. The system of Claim 52, wherein the web server system is operative to cause the browser to execute the page update handler by sending to the browser, with the service data, a command that instructs the browser to execute the page update handler.

59. The system of Claim 52, wherein the web server system is operative to selectively invoke said process based on a programmatic determination that is dependent upon a response time of said service.

60. The system of Claim 52, wherein the web server system comprises computer storage that stores the page update handler, and the web server system is operative to send the page update handler to the browser.

61. The system of claim 60, wherein the page update handler comprises browser-executable code for incorporating the service data into the first portion of the web page as mouse-over text that is displayed by the browser when a mouse cursor is positioned over a corresponding display element.

62. The system of claim 60, wherein the page update handler comprises browser-executable code for adding format coding to the service data to incorporate the viewable representation of the service data into the web page.

63. The system of claim 60, wherein the page update handler comprises executable code for selecting, based at least in part on a dimension of a window of the browser, a display format to use to display the service data in the web page.

64. The system of claim 60, wherein the page update handler comprises JavaScript code.

65. The system of claim 52, wherein the web server system is additionally programmed to respond to the request from the browser by dynamically generating the first portion of the web page using data retrieved by the web server system from at least one additional service.

66. The system of Claim 65, wherein the web server system is operative to incorporate into said first portion of the web page a viewable status message reflecting that additional data is being retrieved.

67. The method of claim 1, wherein the method additionally comprises responding to the page request by dynamically generating the first portion of the web page on said server using data retrieved from at least one additional service.

68. The method of claim 67, wherein dynamically generating the first portion of the web page comprises incorporating into the web page a viewable status message reflecting that additional data is being retrieved.

Appl. No. : 10/720,712
Filed : November 24, 2003

EVIDENCE APPENDIX

None

Appl. No. : 10/720,712
Filed : November 24, 2003

RELATED PROCEEDINGS APPENDIX

None